

<https://mvpdagen.no>

MVP Dagen 2019

NextGen IT-Pro: Automatisasjon og optimalisering i Microsoft 365 og Azure

 @MVPdagen
 #MVPdagen

Automatiser Microsoft 365 med Microsoft Graph

GET [https://graph.microsoft.com/v1.0/me/?\\$select=displayName,jobTitle,companyName,department](https://graph.microsoft.com/v1.0/me/?$select=displayName,jobTitle,companyName,department)

```
{  
  "displayName": "Jan Vidar Elven",  
  "jobTitle": "Architect",  
  "companyName": "Skill",  
  "department": "MVP Enterprise Mobility"  
}
```

GET [https://graph.microsoft.com/v1.0/me/photo/\\$value](https://graph.microsoft.com/v1.0/me/photo/$value)

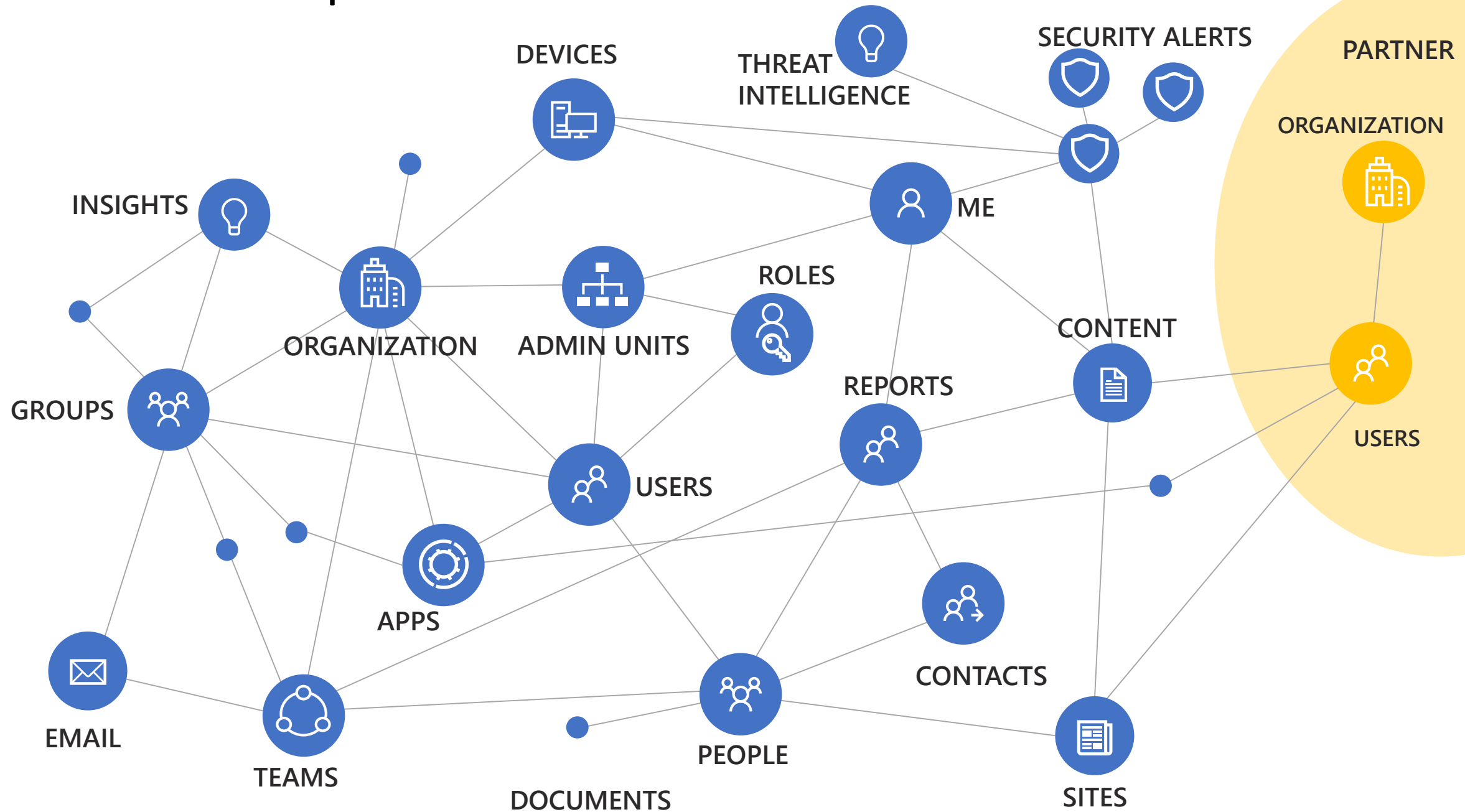


 @JanVidarElven
 @JanVidarElven
 [linkedin.com/in/JanVidarElven](https://www.linkedin.com/in/JanVidarElven)
 gotoguy.blog
 github.com/JanVidarElven

ABOUT MICROSOFT GRAPH

What is it and why is it important?

Microsoft Graph



Microsoft Graph

<https://graph.microsoft.com>

ALL

your data across
Microsoft 365

Office 365
Windows 10
EMS

ALL

types of users

Corporate (@contoso.com)
Consumer (@outlook.com)
Education (@myschool.com)

ONE

way to access it

One endpoint
One auth key
One set of docs
One SDK

GETTING STARTED WITH GRAPH

Microsoft Graph 101

Calling the API

`https://graph.microsoft.com`

`/ {version} / {resource} / {id} / {property} ? {query-parameters}`

- HTTP verbs dictate the request intent: GET | POST | PATCH | PUT | DELETE
- Version: `/v1.0` or `/beta`
- Resource: `/users`, `/groups`, `/sites`, `/drives`, `/devices`, more...
- Member from collection: `/users/jan.vidar@elven.no`
- Property: `/users/jan.vidar@elven.no/department`
- Traverse to related resources via navigations:
`/users/jan.vidar@elven.no/registereddevices`
- Query parameters: `/users/jan.vidar@elven.no/registereddevices?$top=5`
 - Format results: `$select` | `$orderby`
 - Control results: `$filter` | `$expand`
 - Paging: `$top` | `$skip` | `$skiptoken`

YOUR GO TO GUIDE FOR GRAPH LEARNING

- Microsoft Graph Home: <https://graph.microsoft.com>
(<https://aka.ms/Graph>)
- Docs Site for Graph:
<https://docs.microsoft.com/en-us/graph/>
- Graph Explorer: <https://aka.ms/GE>

AUTHENTICATION & PERMISSIONS

Authentication to Microsoft Graph and Permission Scopes

App types and permissions

Get access on behalf of users



Single page app



Web app

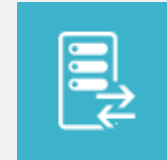


Mobile or
desktop app



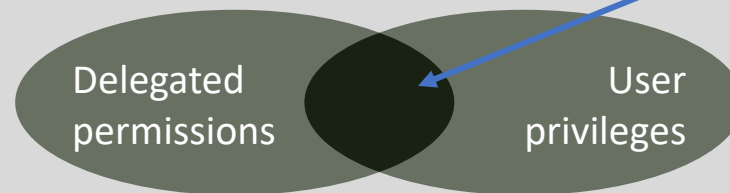
App with
middle tier web API

Get access as a service



Service or
daemon app

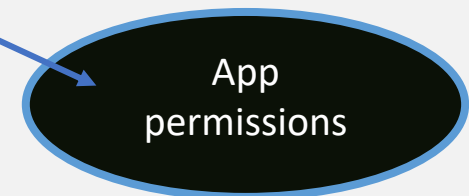
Permission type: delegated



Effective permission

Users can consent for their data or admin can consent for all users

Permission type: application

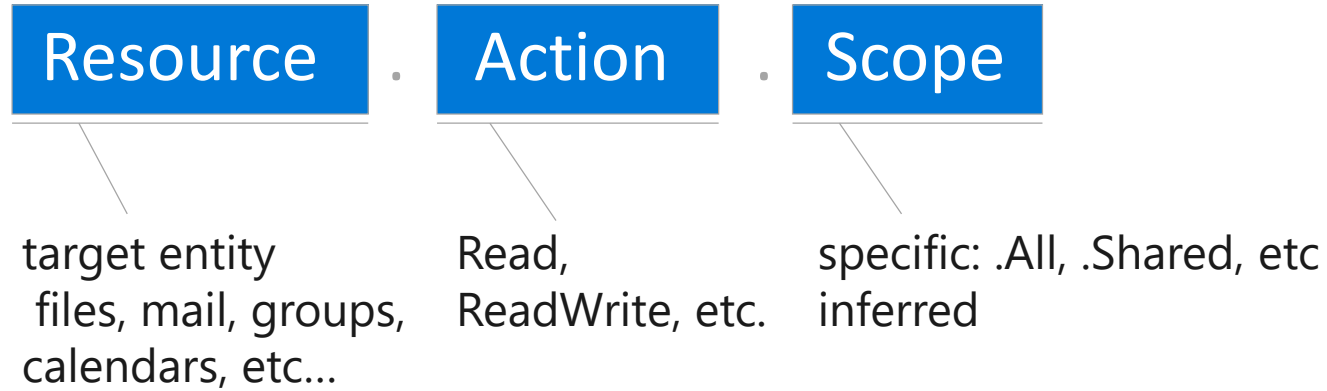


Only admin can consent

Learn more: <https://aka.ms/ConsentAndPermissions>

Permissions

- Microsoft Graph permission names:



- Examples
 - User.Read
 - Notes.ReadWrite
 - Directory.ReadWrite.All

USE CASE MICROSOFT 365 AUTOMATION

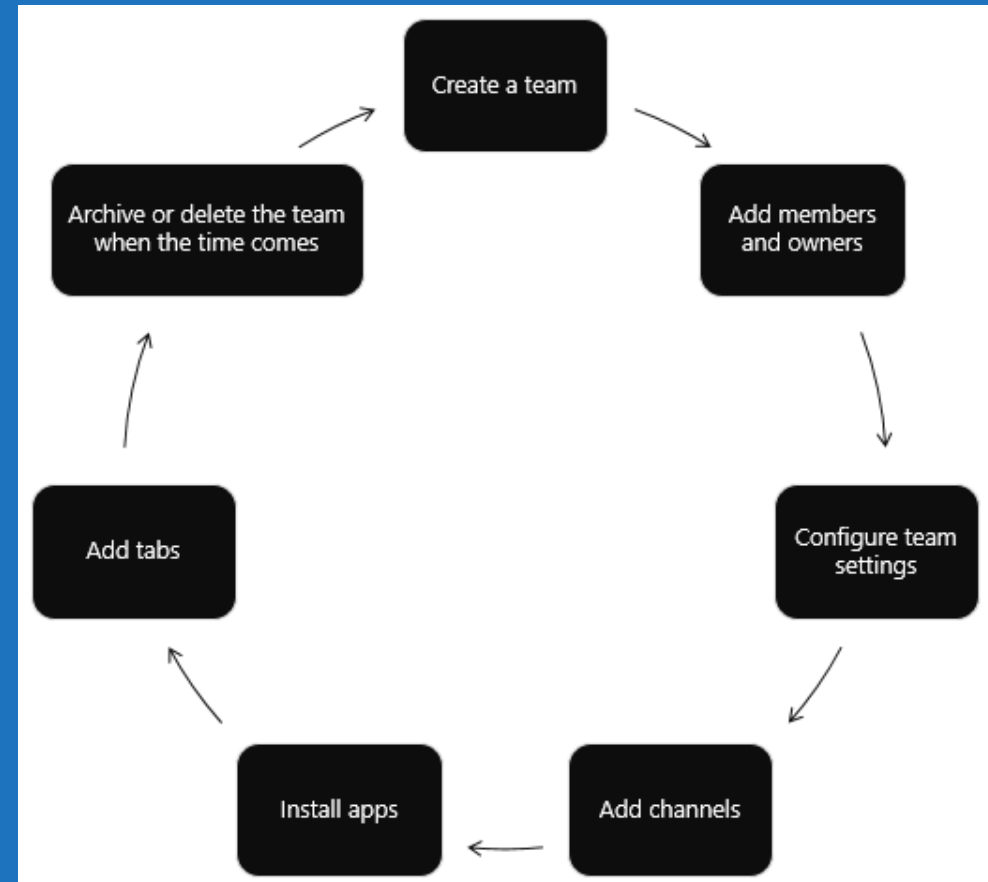
Approach to Use Microsoft Graph for Automating Microsoft 365

FIND YOUR USE CASE FOR M₃65 AUTOMATION

- Repeatable?
- Standardizing IT processes?
- Self Service?
- Approvals?
- Delegation of permissions?

EXAMPLE USE CASE: USE TEAMS FOR OFFICE 365 GOVERNANCE

- <https://techcommunity.microsoft.com/t5/Microsoft-Teams/An-optimal-Guide-to-configure-a-basic-Governance-Team-for-Office/m-p/821660>
- Guide by Magnus Goksøyr [@o365And](#)
- Manual steps for creating a Team with settings, channels and apps



MY STEPS FOR AUTOMATION WITH GRAPH

1. Find manual operations
2. Identify Microsoft Graph API's
3. Identify permissions needed
4. Explore and test Graph requests
5. Explore and choose Automation Platform
6. Publish Solution to (selected) End Users

1. FIND MANUAL OPERATIONS

- Create an empty Team
- Add a custom list to the Office 365 Group for the Team
- Change list settings
- Create a column in the list
- Add the list to a tab in Teams
- (add responsible persons to services)
- Upload Governance plan to Files in Teams
- Make the Governance plan a Tab in Teams
- Create a Channel for Team Requests
- Create a Channel for Change Management
- Create a Tab for Planner for the Change Management Channel, using existing plan
- For each Service Channel, add a Notes tab referencing the Teams OneNote section.
- Rename Tab
- Add Buckets to Plan
- Create a Recurring Meeting
- Create Channel(s) for Office 365 Services (all services you want to governance)
- Add Tab for Roadmap for selected Services channels, linking to Microsoft Roadmap site.
- Upload Service Governance Plan to Files in the channel.
- Make the Governance Plan a Tab in that channel.
- (E-mail alerts for Services alerts)
- Add a Section to the Teams OneNote for each service

2. IDENTIFY GRAPH API'S (DOCS)

The screenshot shows the Microsoft Graph documentation website. The left sidebar contains a navigation menu with categories like 'Sites and lists', 'Social and workplace intelligence', 'Tasks and plans', 'Teamwork', 'List all teams', 'Configuring built-in tabs', 'Protected APIs', 'Send proactive messages with a bot', 'Workbooks and charts', 'Develop', 'Get auth tokens', 'Use the API', 'Migrate', 'Use SDKs', 'Use the toolkit (preview)', 'Resources', 'Best practices', 'Known issues', 'API changelog', and 'Errors'. The 'Teamwork' category is expanded, and 'Create a group and team' is highlighted. The main content area is titled 'Creating a group with a Microsoft Teams team' and includes a sub-header 'Create a group'. It lists two steps: 'Create a group with the right properties' and 'Add a team to the group'. Below this, it provides an example of the JSON payload for a POST request to the /groups endpoint.

Microsoft | Microsoft Graph Solutions ▾ Graph Explorer Getting Started ▾ Docs API Reference Resources ▾ Programs ▾

Learn / Teamwork / Create a group and team [Bookmark](#) [Feedback](#) [Edit](#)

Filter by title

Sites and lists

> Social and workplace intelligence

Tasks and plans

Teamwork

Overview

Create a group and team

List all teams

Configuring built-in tabs

Protected APIs

Send proactive messages with a bot

> Workbooks and charts

Develop

> Get auth tokens

> Use the API

> Migrate

> Use SDKs

> Use the toolkit (preview)

Resources

Best practices

Known issues

API changelog

Errors

Creating a group with a Microsoft Teams team

01/12/2019 • 2 minutes to read •

Creating a [group](#) that includes a [team](#) involves two steps:

- Create a group with the right properties.
- Add a team to the group.

Create a group

In order to include a team, you need to set the following property values, as shown in the following example:

- **groupTypes** = { "Unified" }
- **mailEnabled** = true
- **securityEnabled** = false

HTTP [Copy](#)

```
POST /groups
{
  "displayName": "Flight 157",
  "mailNickname": "flight157",
  "description": "Everything about flight 157",
  "visibility": "Private",
  "groupTypes": ["Unified"],
  "mailEnabled": true,
  "securityEnabled": false,
  "members@odata.bind": [
```



2. IDENTIFY GRAPH API'S

Operation	Microsoft Graph API
Create an empty Team	POST https://graph.microsoft.com/v1.0/groups PUT https://graph.microsoft.com/v1.0/groups/{id}/team OR POST https://graph.microsoft.com/beta/teams
Add a custom list to the Office 365 Group for the Team	POST https://graph.microsoft.com/v1.0/sites/{site-id}/lists
Change list settings	(Settings are defined in Request Body when list is created)
Create a column in the list	(Custom columns are defined in Request Body when list is created)
Add the list to a tab in Teams	POST https://graph.microsoft.com/v1.0/teams/{id}/channels/{id}/tabs
(add responsible persons to services)	POST https://graph.microsoft.com/v1.0/sites/{site-id}/lists/{list-id}/items
Upload Governance plan to Files in Teams	POST https://graph.microsoft.com/v1.0/groups/{groupId}/drive/items/{item-id}/createUploadSession

3. IDENTIFY PERMISSIONS

Request	Delegated Permissions	Application Permissions
POST /groups	Group.ReadWrite.All, Directory.ReadWrite.All, Directory.AccessAsUser.All	Group.ReadWrite.All, Directory.ReadWrite.All
PUT /groups/{id}/team	Group.ReadWrite.All	Group.ReadWrite.All
POST /teams	Group.ReadWrite.All	Group.ReadWrite.All
POST /sites/{site-id}/lists	Sites.Manage.All	Sites.ReadWrite.All
POST /teams/{id}/channels/{id}/tabs	Group.ReadWrite.All	Group.ReadWrite.All
POST /sites/{site-id}/lists/{list-id}/items	Sites.ReadWrite.All	Sites.ReadWrite.All
POST /groups/{groupId}/drive/items/{itemId}/createUploadSession PUT <uploadUrl>	Files.ReadWrite, Files.ReadWrite.All, Sites.ReadWrite.All	Sites.ReadWrite.All
POST	Group.ReadWrite.All	Group.ReadWrite.All

3. ADD PERMISSIONS TO APP

MSGraph Automating Governance Team - API permissions

Search (Ctrl+ /)

Overview

Quickstart

Manage

Branding

Authentication

Certificates & secrets

API permissions

Expose an API

Owners

Roles and administrators (Previ...

Manifest

API permissions

Applications are authorized to call APIs when they are granted permissions by users/admins as part of the consent process. Th configured permissions should include all the permissions the application needs.

+ Add a permission

API / PERMISSIONS NAME	TYPE	DESCRIPTION	ADMIN CONSENT RE...	STATUS
▼ Microsoft Graph (6)				
Files.ReadWrite.All	Delegated	Have full access to all files user c...	-	✓ Granted
Group.ReadWrite.All	Delegated	Read and write all groups	Yes	✓ Granted
Notes.ReadWrite.All	Delegated	Read and write all OneNote note...	-	✓ Granted
Sites.Manage.All	Delegated	Create, edit, and delete items an...	-	✓ Granted
Sites.ReadWrite.All	Delegated	Edit or delete items in all site col...	-	✓ Granted
User.Read	Delegated	Sign in and read user profile	-	✓ Granted

4. EXPLORE & TEST GRAPH REQUESTS

- My Favorite Tools for exploring and testing:

- **Graph Explorer**

- <https://aka.ms/GE>

- **Postman**

- <https://getpostman.com>
 - <https://github.com/microsoftgraph/microsoftgraph-postman-collections>
 - <https://docs.microsoft.com/en-us/graph/use-postman>

- **Visual Studio Code**

- <https://code.visualstudio.com>



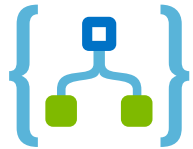
5. EXPLORE AND CHOOSE AUTOMATION PLATFORM



- PowerShell



- Azure Automation



- Azure Logic Apps



- Microsoft Flow



- Azure Functions

- OR

- ..Build your own App/Bot/..

6. PUBLISH & SHARE SOLUTION

- Depending on Automation Platform chosen
- Collaborate & Build on Azure DevOps
- Create Schedules, Triggers
- Integrate with Events and Alerts
- Integrate with Teams
- Integrate with Forms

RESOURCES & QA

- <https://aka.ms/30DaysMSGraph>
- <https://gotoguy.blog>



MVP Dagen 2019

NextGen IT-Pro: Automatisasjon og optimalisering i Microsoft 365 og Azure

Tusen Takk for oss!

 @MVPdagen
 #MVPdagen

<http://mvpdagen.no>